This is the accepted version of a paper presented at International Conference on Humanoid Robots (Humanoids), Cancun, 15-17 Nov. 2016.

Citation for the original published paper:

Almeida, D., Viña, F E., Karayiannidis, Y. (2016)
Bimanual Folding Assembly: Switched Control and Contact Point Estimation.
In: IEEE-RAS 16th International Conference on Humanoid Robots (Humanoids), Cancun, 2016 Cancun: IEEE
https://doi.org/10.1109/HUMANOIDS.2016.7803279

N.B. When citing this work, cite the original published paper.

Permanent link to this version:
http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-199856
Bimanual Folding Assembly: Switched Control and Contact Point Estimation

Diogo Almeida  Francisco E. Viña  Yiannis Karayiannidis

Abstract—Robotic assembly in unstructured environments is a challenging task, due to the added uncertainties. These can be mitigated through the employment of assembly systems, which offer a modular approach to the assembly problem via the conjunction of primitives. In this paper, we use a dual-arm manipulator in order to execute a folding assembly primitive. When executing a folding primitive, two parts are brought into rigid contact and posteriorly translated and rotated. A switched controller is employed in order to ensure that the relative motion of the parts follows the desired model, while regulating the contact forces. The control is complemented with an estimator based on a Kalman filter, which tracks the contact point between parts based on force and torque measurements. Experimental results are provided, and the effectiveness of the control and contact point estimation is shown.

I. INTRODUCTION

Robotic manufacturing is a current challenge for the scientific community. While successful in the automotive sector, it has not been deployed for automating small scale assembly, or the assembly of electronic devices. This is in part due to the fact that robots are still not ready to operate in unstructured environments, where human interference is present either as a disturbance or as a collaborative partner on the task itself and not all the relevant task parameters are known with a high degree of certainty.

In our previous work we introduced folding assembly for a dual-arm manipulator as an alternative approach to some common assembly steps. It does not rely on models of the assembly parts, using force and torque measurements to handle the uncertainty on the task. Equipping a robotic assembly system with a folding motion primitive allows it to increase the available options when facing this kind of tasks, adding robustness to the system as a whole. We showcased early results of executing folding with a simple feedback controller and a heuristic contact point computation between parts, as a proof of concept [1].

In this paper, we improve the folding assembly execution by introducing a contact point estimator based on a Kalman filter (section V). By using the kineto-statics model for the assembly task with rigid grasps (section III), process and measurement models arise naturally. Furthermore, we expand on the control by taking into account the switching nature of the system, that can have a contact point that is either allowed to translate, or limited to pure rotations (section IV).

Experimental results are shown in section VI. We demonstrate the contact point estimation, on a dual-arm manipulator, on a master-slave setup using the switched controller, where one arm acts as a passive sensor and the other actuates on one of the parts.

The main contributions of this paper are as follows:

• Introduction of a switched control strategy for the folding assembly task. The interaction between parts is controlled taking into account the type of contact
• Improvements on the contact point estimation by means of Kalman filtering. Previous work used a direct computation based on the norms of the force and torque signal, which did not take into account either the model of the problem or the noise affecting the measurements
• Experimental analysis of the estimation and control execution.

II. RELATED WORK

Large scale industrial assembly lines have been a successful example of how robotics can be used in order to automate assembly tasks. The automotive industry is a sector with several automated assembly steps. Taking advantage of the average available time to setup an automotive assembly line, robots that have few to none environmental uncertainties can execute laborious assembly steps [2]. The average product life cycle for this particular example helps in determining the success of robotic manipulators with some industrial agents [3].
Presently, small scale assembly is a challenging task to automate. The life cycle of products such as cellphones and other electronic devices is significantly shorter than for motor vehicles, which leads to the need for quickly setting up an assembly line. This is in conflict with the implementation complexity of a carefully calibrated robot [4]. Thus, in order to achieve a reduced assembly line setup time, there is the need for robotic systems to work robustly in unstructured environments. Without using a large amount of adaptation time, robots should be able to execute a given task with uncertainties in variables such as the positioning of assembly parts. This is in contrast with traditional robotic assembly, where most if not all variables of interest are known with a high degree of precision.

Manipulation and assembly systems are built around the concept of primitives, that can be composed in order to generate a complex behaviour. These are used so that the act of programming a robot becomes simpler [5], and to increase the level of abstraction when creating such a system [6]. In [7], a manipulation system is able to execute different grasps through the conjunction of motion primitives, motion planning and low level arm controllers. The system [8] takes advantage of the modularity of movement primitives in order to execute concurrent motions in a coordinated fashion, while [9] develops a hierarchical classification of manipulation primitives.

There are some variations on the definition of assembly/motion primitive. In [10], “Robotic Assembly Skills” are defined with the purpose of establishing a clear interface with higher level logic. Other, more formal, definition can be found in [6], with the aim of developing a manipulation primitives based control system. Finally, a graphical execution model [11] uses an assembly primitive definition that helps in automating error recovery during the execution of a complex assembly task. All of the definitions have in common the notion that an assembly primitive should execute one well defined subtask.

Robotic perception is a challenge when trying to accomplish an assembly task. While occlusion may occur during task execution, camera images, as well as point cloud data, can be used to gather data on objects prior to manipulation, which allows for the robot to plan an action. This is the case in the sensing of articulate objects, where the kinematic properties of objects can be estimated using vision, to aid in the manipulation task. Vision alone is usually not sufficient for effective manipulation, and interactive perception techniques may be used to improve results [12], [13].

During the execution of an assembly task, occlusion and limited precision of the vision system may have a negative impact on the performance of the robot, which can be mitigated via the employment of force and torque sensors. A peg-in-hole task is a classical example that can be solved through a manipulation primitive. Force sensing can be used in order to compensate for uncertainties in the peg orientation with respect to the hole [14], and it allows the system to operate without explicit knowledge of the dimensions of the grasped objects [15]. More recently, the problem of snap-fitting has been subject of attention. A manipulator is supposed to assemble an object that requires the application of a significant amount of force in order to ‘snap’ the parts together [16], [10].

Our work deals with the manipulation of rigid objects that, when brought into contact, form a virtual articulated object. By monitoring force and torque measurements at the robot wrist, we circumvent the need for visually tracking the parts during the task execution. The estimation of contact dynamics between a rigidly grasped object and a compliant environment is done in [17], where a Kalman filter was employed to get the contact parameters. This includes not only the contact point between a probe and the robotic end-effector, but also an estimate of the friction and a measure of the environment compliance. Alternatively, the contact point estimation problem has been solved through the employment of adaptive estimation techniques [18].

For problems where multiple contact hypothesis may exist, as is the case when trying to find the hole in a peg in hole assembly scenario, for example, particle filtering is a viable alternative approach, taking into consideration the added computational complexity [19], [20].

III. THE FOLDING ASSEMBLY TASK

The main motivation for implementing a folding assembly controller in a robotic system is to equip it with a readily available tool that allows for a generic translation and rotation of two parts in contact. Some examples of this are the assembly of a cellphone (Fig. 2) or the insertion of a battery in a remote control. In this section, we summarize part of the folding assembly model introduced in [1].
Consider two robotic manipulators, applying rigid grasps on two rigid parts. The purpose of a folding assembly step is to, upon bringing the parts into rigid contact, perform a controlled motion of the contact point between them. This implies a regulation of the contact forces between parts while executing the task. Furthermore, the contact point cannot be measured directly, and must be estimated.

We name the two parts according to their semantic role in the assembly step: the surface piece will be interpreted as the part that would be kept in place by a fixture in a single-arm assembly scenario. It should usually be the part that accommodates the other piece after the successful execution of the assembly step. The rod piece is the part that would be manipulated in a single-arm scenario, that is expected to be hosted by the surface piece after the successful execution of the assembly step. We model the motion of the independent pieces as that of an articulated object with a single prismatic revolute joint connecting two rigid bodies.

A. Kineto-statics

For this work, we focus on the assumption that both assembly parts are being grasped rigidly, with the manipulators being able to perform independent motion. We also assume a purely planar operation. This is a reasonable assumption as long as lateral movement of the assembly parts is constrained, and a rough estimate of the desired operation plane is provided. The notation used in this article is as follows:

- Lower case letters in bold denote vectors, while bold upper case letters are used for matrices. The transpose of a vector or matrix is denoted as \( \boldsymbol{v}^\top \).
- The matrix \( \mathbf{S}(\boldsymbol{\omega}) \) is the skew-symmetric matrix,

\[
\mathbf{S}(\boldsymbol{\omega}) = \begin{bmatrix}
0 & -\omega_z & \omega_y \\
\omega_z & 0 & -\omega_x \\
-\omega_y & \omega_x & 0
\end{bmatrix}.
\]

The skew-symmetric matrix is used to express the cross product, such that \( \boldsymbol{\omega} \times \boldsymbol{x} = \mathbf{S}(\boldsymbol{\omega})\boldsymbol{x} \).

- The surface piece normal and tangent vector are denoted respectively by \( \mathbf{n} \) and \( \mathbf{t} \).

We use the indices \( j \in \{1, 2\} \) to denote respectively the robotic end-effectors that grasp the rod and the surface piece, with \( \mathbf{p}_j \) indicating their coordinates in the world frame. We also define \( \mathbf{p}_c \) as the position of the contact point between the parts in the world frame. Let us define the vector \( \mathbf{\theta}_c \triangleq \theta_c \mathbf{t} \times \mathbf{n} \) where \( \theta_c \) is the angle between rod and the surface piece, and \( \mathbf{t} \times \mathbf{n} \) is the rotation axis. Finally, the frames \( \{h_1\} \) and \( \{h_2\} \) are used at the two respective grasping points, with \( \{h_{c1}\} \) and \( \{h_{c2}\} \) being located at the contact point, but with the same orientation of the respective grasping point frames, such that \( \{h_1\} \) has the same orientation as \( \{h_{c1}\} \) and \( \{h_2\} \) is oriented as \( \{h_{c2}\} \) (Fig. 4).

With these definitions, the rigid grasp assumption is translated as a constraint on the frames angular velocities, denoted as \( \boldsymbol{\omega}_j = \mathbf{S}(\omega_{ij}) \), and the linear velocities of the end effectors, \( \mathbf{v}_j \), can be written in terms of the velocity of the contact point, \( \mathbf{v}_c \) and the vectors \( \mathbf{r}_j = \mathbf{p}_c - \mathbf{p}_j \) connecting the contact point \( \mathbf{p}_c \) to each grasping point \( \mathbf{p}_j \):

\[
\begin{align*}
\mathbf{v}_1 &= \mathbf{v}_c - \mathbf{S}(\omega_1)\mathbf{r}_1 \\
\mathbf{v}_2 &= \mathbf{v}_c - \mathbf{S}(\omega_2)\mathbf{r}_2,
\end{align*}
\]

where \( \mathbf{v}_2 \parallel \) is the component of \( \mathbf{v}_2 \) along the surface piece tangent. The reaction wrench, \( \mathbf{w}_j = [\mathbf{f}_j^\top, \mathbf{\tau}_j^\top]^\top \in \mathbb{R}^6 \), is then defined along the reciprocal of the movement directions:

\[
[\mathbf{v}_j^\top \mathbf{\omega}_j^\top] \mathbf{w}_j = 0. 
\tag{1}
\]

During a folding assembly execution with rigid grasps on both pieces, we have two possible cases for the expected reaction forces: the contact point is either able to slide along the surface piece, or is fixed in place:

1) Sliding contact point: When sliding, the contact point behaves as a prismatic revolute joint (Fig. 5a), which allows for forces to be exerted only along the surface piece normal, this imposes the constraint \( \mathbf{n}^\top \mathbf{v}_c = 0 \), and using equation (1) we obtain

\[
\mathbf{w}_j = \begin{bmatrix} \mathbf{n}^\top \mathbf{S}(\mathbf{r}_j) \mathbf{n} \end{bmatrix} \lambda_j, 
\tag{2}
\]

where \( \lambda_j \in \mathbb{R} \) are Lagrange multipliers [21], in Newtons, N.
2) Fixed contact point: The fixed contact point will only be able to rotate around its position (Fig. 5b), acting as a pure revolute joint. The reaction forces are

\[ w_j = \begin{bmatrix} I \\ S(r_j) \end{bmatrix} \lambda_j, \]

with \( \lambda_j \in \mathbb{R}^3 \).

IV. CONTROL OF A FOLDING ASSEMBLY TASK

Assuming that the parts begin in contact, the purpose of our controller is to: a) move the contact point in order to achieve an assembly goal and b) regulate the contact forces during the assembly. The contact point kinematics are given by:

\[ v_c = S(\omega_1)r_1 + v_1 \]

\[ \omega_c = \omega_1, \]

that corresponds to the model of the virtual prismatic revolute joint. We use the feedback linearising control strategy from [1]:

\[ v_1 = -S(\omega_1)(p_c - p_1) + v_{\text{ref}}, \]

\[ \omega_1 = \omega_{\text{ref}}. \]

That is, we compensate for the contact point kinematics while introducing a reference twist \( u = [v_{\text{ref}}^T, \omega_{\text{ref}}^T]^T \). Due to the nature of the problem, the reference velocity \( v_{\text{ref}} \) may be composed of a motion and force compensation components, or be exclusively the product of a force regulation component.

A. Motion and force control

We make the control signal \( u \) a function of the contact point and a discrete task state \( q \in Q \), with the set \( Q = \{ q_1, q_2 \} \), that is, \( u = f(p_c, \theta_c, q) \), where \( \theta_c = \theta_c(t) \times n \) is the smallest angle formed between the two parts in contact, as in Fig. 5. We use two distinct control structures for each state (Fig. 6).

1) Sliding contact point - State \( q_1 \): In this state, forces are exerted along the surface normal. We have thus two kinematic degrees of freedom to be commanded, a translation and rotation, and a force controlled component. Let \( v_d \) be the desired translational velocity of the contact point and let \( \theta_{d1} \) be the desired angle between parts. We define the normal force error as \( f_{n} = f_n \cdot n - f_{\text{ref},n} \), which is regulated by a signal \( v_{f_n} \) defined along the surface normal. Finally, the orientation error is given by \( \theta_{e1} = \theta_c - \theta_{d1} \). The reference twist applied to (5) becomes:

\[ f(p_c, \theta_c, q_1) = \begin{bmatrix} v_d + v_{f_n} \\ \omega_{Q1} \end{bmatrix} \]

with

\[ v_{f_n} = -K_{f_n} \int f_{n} n \, dt - K_{p_n} f_{n} n \]

\[ \omega_{Q1} = -K_w \theta_{e1}. \]

The transitions between states are defined by a switching rule \( \sigma(f_{||}, q) \), where \( f_{||} = f_{||}^T t \). When \( f_{||} \) overcomes a certain threshold, we assume that the contact point is fixed, and the control switches to state \( q_2 \).

\[ \sigma(f_{||}, q_1) = \begin{cases} q_1, & \| f_{||} \| < f_b \\ q_2, & \| f_{||} \| > f_b \end{cases} \]

2) Fixed contact point - State \( q_2 \): In the fixed contact point state we only have one kinematic degree of freedom, the rotation around the current contact point. This implies two independent directions on which to exert forces and, thus, we define a tangential force error, \( f_{t} = f_t \cdot t - f_{\text{ref},t} \), and a second orientation error \( \theta_{e2} = \theta_c - \theta_{d2} \). Denoting as \( v_{f_t} \) the velocity component that regulates the tangential force error, we apply the following reference twist:

\[ f(p_c, \theta_c, q_2) = \begin{bmatrix} v_{f_t} + v_{f_t} \\ \omega_{Q2} \end{bmatrix} \]

where

\[ v_{f_t} = K_{f_t} \int f_{t} t \, dt + K_{p_t} (f_{t} t) \]

\[ \omega_{Q2} = -K_w \theta_{e2}. \]

Should \( \| f_{||} \| \) decrease below a threshold (if, for instance, the contact point was only apparently fixed due to irregularities in the contact surface), the control switches back to \( q_1 \),

\[ \sigma(f_{||}, q_2) = \begin{cases} q_1, & \| f_{||} \| < f_a \\ q_2, & \| f_{||} \| > f_a \end{cases} \]

we make the transition thresholds \( 0 < f_a < f_b \) in order to allow for hysteresis and thus prevent high frequency switching due to sensor noise.

V. CONTACT POINT ESTIMATION

When executing the folding controller, the contact point is required in order to compensate for the kinematics in equation (5) and to compute the angle between rod and surface piece. We can write (4) in terms of \( p_c \),

\[ p_c = S(\omega_1)(p_c - p_1) + v_1. \]

We will write (8) discretely, and derive the Kalman filter process and observation models from it, using the notation
$$\text{TABLE I: Controller and estimator parameters}$$

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$K_{I_n}$</td>
<td>$5 \times 10^{-4}$</td>
</tr>
<tr>
<td>$K_{P_n}$</td>
<td>$25 \times 10^{-9}$</td>
</tr>
<tr>
<td>$K_{I_t}$</td>
<td>$10^{-4}$</td>
</tr>
<tr>
<td>$K_{P_t}$</td>
<td>$25 \times 10^{-5}$</td>
</tr>
<tr>
<td>$K_{\omega}$</td>
<td>$5 \times 10^{-2}$</td>
</tr>
<tr>
<td>$\Delta_t$</td>
<td>$3 \times 10^{-3}$ s</td>
</tr>
<tr>
<td>$R$</td>
<td>$5 \times 10^{-2} \text{I}$</td>
</tr>
<tr>
<td>$Q$</td>
<td>$25 \times 10^{-2} \text{I}$</td>
</tr>
</tbody>
</table>

from [22]. If we take the contact point at time $t = k$, (8) becomes

$$p_{c_k} = p_{c_{k-1}} + (S(\omega_{1_k})(p_{k-1} - p_{1_k}) + v_1) \Delta_t. \quad (9)$$

where $\Delta_t$ is the sampling period of the controller. Let $\hat{p}_{c_k}$ denote the contact point estimate at time $t = k$. The process model of the Kalman filter is written as

$$\hat{p}_{c_k} = A_k \hat{p}_{c_{k-1}} + B_k u_k + c_k, \quad (10)$$

and the model parameters can be inferred directly from (9):

$$A_k = I + S(\omega_{1_k}) \Delta_t \quad B_k = I \Delta_t \quad u_k = v_{1_k} \quad c_k = -S(\omega_{1_k}) p_{1_k} \Delta_t,$$

Additive gaussian white noise is assumed, which is parametrized by a covariance matrix $R$.

Given the rigid grasp assumption, we have that both models (2) and (3) have the same relationship between forces and torques. In discrete time:

$$\tau_{j_k} = S(r_{j_k}) f_{j_k} \quad (11)$$

And we can then use the relationship (11) and the observation model for the Kalman filter,

$$z_k = C_k \hat{p}_{c_k}, \quad (12)$$

with

$$z_k = \tau_2 - S(f_2) p_2 \quad C_k = -S(f_2).$$

We use $j = 2$ since the force and torque measurements were taken from the surface piece end-effector. We recall that the subscript $j = 1$ indicates the rod piece end-effector, which actuates on the moving part, whereas $j = 2$ denotes the surface piece end-effector, which remains static during the contact activity.

The observations are assumed to be perturbed by additive gaussian white noise, parametrized by a covariance matrix $Q$. The surface piece end-effector, being static, will not be affected by inertial effects, which improves the model (12) reliability over what could be expected from using the rod piece end-effector. The filter maintains an estimate in terms of the first two gaussian moments, the mean $\mu$ and a covariance $\Sigma$, which are updated according to the known equations:

$$\mu_k = A_k \mu_{k-1} + B_k u_k + c_k \quad \Sigma_k = A_k \Sigma_{k-1} A_k^\top + R \quad K_k = \Sigma_k C_k^\top (C_k \Sigma_k C_k^\top + Q)^{-1} \quad \mu_k = \mu_k + K_k (z_k - C_k \hat{\mu}_k) \quad \Sigma_k = (I - K_k C_k) \Sigma_k. \quad (13)$$

It follows that the estimate of the contact point is given by $\hat{p}_k = \mu_k$. Note that, when the contact point is not sliding, the model (10) should be equivalent to a random walk model, that is, the different components of the process model should cancel themselves out to ensure $\mu_k = \mu_{k-1}$.

VI. EXPERIMENTAL RESULTS

The contact point estimation and folding controller were experimentally tested on the platform described in [23]. It
is a dual-arm robot equipped with two 7 DOF manipulators, with force torque sensors at the wrists. An experiment is initialized with the robot grasping two 3D printed rigid parts, Fig. 3, which are brought together with a specified contact force. The estimator is initialized with $\mu = p_2, \Sigma = 10^{-2}I$. This can lead to an initial jerky motion of the rod piece manipulator, depending on the actual initial contact state. As a pre-processing step, the controller is iterated 100 times (0.3 s) with the available for and torque data, with no actuation. This minimizes aggressive transients.

The controller initial state is set as $q_1$. Control gains and estimator parameters are set as in table I. While the exact value of these parameters is the product of fine tuning based on the actual performance of the robot, it is important to note that the noise covariance matrix should account for the usually noisy data obtained from force and torque sensors. In this case, this mean that it should be two or three orders of magnitude bigger than the process noise covariance matrix (the noise on the robot motion is considerably larger than the noise of its force and torque sensors).

The control reference values and state transition forces are shown in table II. The robot does not provide direct measurements of the joint velocities, thus, the velocities used in the process model (4) are approximate, under the assumption that the velocity controller is able to execute the commanded velocities. This implies that the velocity components derived from equations (6) and (7) are subtracted from the commanded velocity $v_1$ before being used in the estimator.

Results from the control execution are depicted in Fig. 7. The controller moves the rod piece along the surface piece normal for approximately 3 seconds, when the tangential forces increase above $f_b$, and it switches to $q_2$. The tangential direction becomes force controlled from there on. The estimated contact point closely tracks the ground truth data until the switch to $q_2$. At this point, the impact on the surface piece fixture deviates the estimate for approximately 2 millimetres with respect to the ground truth. This can be explained by the fact that the grasp on the rod piece is not perfectly rigid, and the employed robotic gripper (Fig. 1) has some compliance: a slight movement with respect to the gripper occurs upon impact, while the ground truth is set as a rigid transform with respect to the rod piece end-effector, the estimator performance is exemplified in Fig. 8. This behaviour is consistent over several experiments.

The assembly task was correctly executed, with the contact point sliding until it hits the surface piece fixture. The rod piece was correctly oriented with respect to the surface piece during the assembly. However, the time span of the whole task had to be made large, given that the platform has a relatively low control frequency, and increasing the desired velocities would amplify oscillations of the robot base, which are captured by the force and torque sensors.

### TABLE II: Controller reference values

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>$v_d$</td>
<td>$v_d$ 1</td>
</tr>
<tr>
<td>$v_d$</td>
<td>5 mm/s</td>
</tr>
<tr>
<td>$\theta_{d1}$</td>
<td>$\theta_{d1}$ 1.0 rad</td>
</tr>
<tr>
<td>$\theta_{d2}$</td>
<td>$\theta_{d2}$ -0.9 rad</td>
</tr>
<tr>
<td>$f_a$</td>
<td>2 N</td>
</tr>
<tr>
<td>$f_b$</td>
<td>4 N</td>
</tr>
<tr>
<td>$f_{tot}$</td>
<td>10 N</td>
</tr>
<tr>
<td>$f_{tot}$</td>
<td>5 N</td>
</tr>
</tbody>
</table>

VII. Conclusion

In this paper, we expand on a folding assembly task execution by implementing a switched controller to execute a folding assembly task, while estimating the contact between parts by means of Kalman filtering. We take advantage of the dual-arm platform by taking the force and torque measurements from the static surface piece gripper. This minimizes inertial and Coriolis effects in the estimation, improving the estimate with respect to the direct computation based on the force and torque measurements norm. The addition of PI force controllers along the two possible force dimensions ensured that the forces do not fluctuate upon contact, which was the case in our previous work.

The main advantage of the proposed approach lies in its independence from specific knowledge of the grasped parts, namely its dimensions and geometry. Contact between parts and a rough estimate of the sliding direction is enough to execute the task.

While formulated as a planar contact point problem, the actual task execution involves a contact surface, which helps

in ensuring that the reaction forces are sufficient in order for the estimation equations (13) to converge. Given the constrained nature of an assembly operation, this could give insights in how should the controller behave in order to ensure a correct identification of the contact point kinematics, or, alternatively, what kind of contact should the parts have for a successful estimation.

Finally, the application of compliant force controllers to this problem would allow for the forces to remain inside predetermined ranges, while possibly minimizing the required control effort on the robot part.

REFERENCES


